What is claimed is:

Sub A1

1. A program converting unit for generating a machine language instruction from a source program for a processor that manages an N-bit address while processing M-bit data, N being greater than M, said program converting unit comprising:

parameter holding means for holding a data width and a pointer width designated by a user, said data width representing the number of bits of data used in the source program while said pointer width representing the number of bits of an address; and

generating means for generating an instruction to manage said data width when a variable operated by said instruction represents the data, and for generating an instruction to manage said pointer width when a variable operated by said instruction represents the address.

2. The program converting unit of Claim 1, wherein said M is 16 and said N is an integer in a range of 17 to 31 inclusive.

3. The program converting unit of Claim 1, wherein said generating means includes:

~~judging~~ determining means for ~~judging~~ determining a kind of the machine language instruction, the machine language instruction including (1) an instruction to access to a memory, (2) an instruction to use a register, and (3) an instruction to use an immediate;

memory managing means for outputting a direction, in case of the (1) instruction, to manage said data width as an effective memory-access width when a variable to be accessed represents the data, and to manage said pointer width as an effective memory-access width when said variable represents the address;

register managing means for outputting a direction, in case of the (2) instruction, to manage said data width as an effective bit-width when a variable to be read/written from/into the register represents the data, and to manage said pointer width as the effective bit-width when said variable represents the address;

immediate managing means for outputting a direction, in case of the (3) instruction, to manage said data width as the effective bit-width when said immediate represents the data, and to manage said pointer width as the effective bit-width when said immediate represents the address; and

code generating means for generating the machine language instruction in accordance with the directions from said memory managing means, said register managing means, and said immediate managing means.

4. The program converting unit of Claim 3, wherein said M is 16 and said N is an integer in a range of 17 to 31 inclusive.

5. The program converting unit of Claim 4, wherein:

said N is 24; and

said code generating means generates an instruction for a 24-bit data operation when said pointer width is greater than 16 bits and less than 24 bits, and generates an instruction for a 16-bit data operation when said pointer width is 16 bits or less.

~~6. A program converting unit for generating a machine language instruction based on a source program for a processor that manages an N-bit address while processing M-bit data, N being greater than M, said program converting unit comprising:~~

~~syntax analyzing means for analyzing a syntax of the source program to convert the same into an intermediary language comprising intermediary instructions, and subsequently for judging whether or not each variable contained in said intermediary instructions represents data used in an address;~~

~~table generating means for generating a table for each variable in said intermediary instructions, said table holding a name together with a type of each variable, said type representing one of the data and the address;~~

~~parameter holding means for holding a data width and a pointer width designated by a user, said data width representing the number of bits of the data while said pointer width representing the number of bits of the address; and~~

~~generating means for generating an instruction to manage~~

said data width when the variable in said intermediary instruction represents the data, and an instruction to manage said pointer width when said variable represents the address.

7. The program converting unit of Claim 6, wherein said M is 16 and said N is an integer in a range of 17 to 31 inclusive.

8. The program converting unit of Claim 6, wherein said generating means includes:

judging means for judging a kind of the machine language instruction, the machine language instruction including (1) an instruction to access to an memory, (2) an instruction to use a register, and (3) an instruction to use an immediate;

memory managing means for outputting a direction, in case of the (1) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective memory-access width depending on the type of a variable to be accessed shown in said table;

register managing means for outputting a direction, in case of the (2) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective bit-width depending on the type of a variable to be read/written from/in the register shown in said table;

immediate managing means for outputting a direction, in case of the (3) instruction, to manage a corresponding bit-width held in said parameter holding means for the immediate as an effective bit-width depending on the type of the immediate shown in said table; and

code generating means for generating the machine language instruction in accordance with the directions from said memory managing means, said register managing means, and said immediate managing means.

9. The program converting unit of Claim 8, said M is 16 and said N is an integer in a range of 17 to 31 inclusive.

10. The program converting unit of Claim 9, wherein:

said N is 24; and

said code generating means generates an instruction for a 24-bit data operation when said pointer width is greater than 16 bits and less than 24 bits, and generates an instruction for a 16-bit data operation when said pointer width is 16 bits or less.

11. A program converting unit for generating a machine language instruction based on a source program for a processor that manages an N-bit address while processing M-bit data, N being greater than M, said program converting unit comprising:

syntax analyzing means for analyzing a syntax of the source program to convert the same into an intermediary language comprising intermediary instructions, and subsequently for judging whether or not each variable contained in said intermediary instructions represents data used in an address;

table generating means for generating a table for each variable in said intermediary instructions, said table holding a name together with a type of each variable, said type representing one of the data and the address;

parameter holding means for holding a data width and a pointer width designated by a user, said data width representing the number of bits of the data while said pointer width representing the number of bits of the address;

judging means for judging a kind of the machine language instruction, the machine language instruction including (1) an instruction to access to an memory, (2) an instruction to use a register, and (3) an instruction to use an immediate;

memory managing means for outputting a direction, in case of the (1) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective memory-access width depending on the type of a variable to be accessed shown in said table;

register managing means for outputting a direction, in case of the (2) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective bit-width depending on the type of a variable to be read/written from/in the register shown in said table;

immediate managing means for outputting a direction, in case of the (3) instruction, to manage a corresponding bit-width held in said parameter holding means for the immediate as an effective bit-width depending on the type of the immediate shown in said table; and

code generating means for generating the machine language instruction in accordance with the directions from said memory managing means, said register managing means, and said immediate managing means.

12. The program converting unit of Claim 11, wherein

said code generating means generates an instruction for a 24-bit data operation when said pointer width is greater than 16 bits and less than 24 bits, and generates an instruction for a 16-bit data operation when said pointer width is 16 bits or less.

Sub A0

13. A program converting unit for generating a machine language instruction from a source program for a processor that manages an N-bit address while processing M-bit data, N being greater than M, said program converting unit comprising:

parameter holding means for holding a data width and a pointer width designated by a user, said data width representing the number of bits of data used in the source program while said pointer width representing the number of bits of an address;

generating means for generating an instruction to manage

106

said data width when a variable operated by said instruction represents the data, and for generating an instruction to manage said pointer width when a variable operated by said instruction represents the address;

option directing means for holding a user's direction for an overflow compensation, an overflow being possibly caused by an arithmetic operation; and

compensate instruction generating means for generating a compensation instruction to compensate an overflow in accordance with a type of a variable used in the arithmetic operation, said type being judged when said option directing means holds the user's direction for executing the overflow compensation, said compensation instruction being generated when an effective bit-width of a variable designated by an operand is shorter than a register of N-bit wide and the arithmetic operation instruction will possibly cause an overflow exceeding said effective bit-width.

14. The program converting unit of Claim 13, wherein said M is 16 and said N is an integer in a range of 17 to 31 inclusive.

15. The program converting unit of Claim 13, wherein said M is 32, and said N is an integer in a range of 33 to 63 inclusive.

16. The program converting unit of Claim 13, wherein said compensate instruction generating means includes:

instruction judging means for judging an arithmetic operation instruction that will possibly cause an overflow for all the machine language instructions when said option instructing means holds the user's direction for executing the overflow compensation;

variable judging means, with respect to a variable in the arithmetic operation instruction judged by said instruction judging means, for judging an effective bit-width and whether said variable is signed or unsigned by referring to said table;

sign-extension instruction generating means for generating a compensation instruction in case of a signed variable, a logical value of a sign bit being filled into all bits higher than the effective bit-width in a register that is to store said signed variable by said sign-extension compensation instruction; and

zero-extension instruction generating means for generating a zero-extension compensation instruction in case of an unsigned variable, a logical value "0" being filled into all bits higher than the effective bit width in a register that is to store said unsigned variable by said zero-extension compensation instruction.

17. The program converting unit of Claim 16, wherein said generating means includes:

~~judging~~ determining means for ~~judging~~ determining a kind of the machine language instruction, the machine language instruction including (1) an instruction to access to a memory, (2) an instruction to use a register, and (3) an instruction to use an immediate;

memory managing means for outputting a direction, in case of the (1) instruction, to manage said data width as an effective memory-access width when a variable to be accessed represents the data, and to manage said pointer width as an effective memory-access width when said variable represents the address;

register managing means for outputting a direction, in case of the (2) instruction, to manage said data width as an effective bit-width when a variable to be read/written from/into the register represents the data, and to manage said pointer width as the effective bit-width when said variable represents the address;

immediate managing means for outputting a direction, in case of the (3) instruction, to manage said data width as the effective bit-width when said immediate represents the data, and to manage said pointer width as the effective bit-width when said immediate represents the address; and

code generating means for generating the machine language instruction in accordance with the directions from said memory managing means, said register managing means, and said immediate managing means.

18. The program converting unit of Claim 17, wherein said M is 16 and said N is an integer in a range of 17 to 31 inclusive.

19. The program converting unit of Claim 17, wherein said M is 32, and said N is an integer in a range of 33 to 63 inclusive.

Sub B3

20. A program converting unit for generating a machine language instruction based on a source program for a processor that manages an N-bit address while processing M-bit data, N being greater than M, said program converting unit comprising:

syntax analyzing means for analyzing a syntax of the source program to convert the same into an intermediary language comprising intermediary instructions, and subsequently for judging whether or not each variable contained in said intermediary instructions represents data used in an address;

table generating means for generating a table for each variable in said intermediary instructions, said table holding a name together with a type of each variable, said type representing one of the data and the address, and one of singed and unsigned data;

parameter holding means for holding a data width and a pointer width designated by a user, said data width representing the number of bits of the data while said pointer width representing the number of bits of the address;

option directing means for holding a user's direction for an overflow compensation, an overflow being possibly caused by an arithmetic operation;

generating means for generating an instruction to manage said data width when the variable in said intermediary instruction represents the data, and an instruction to manage said pointer width when said variable represents the address; and

compensate instruction generating means for generating a compensation instruction to compensate an overflow in accordance with a type of a variable used in the arithmetic operation, said type being judged when said option directing means holds the user's direction for executing the overflow compensation, said compensation instruction being generated when an effective bit-width of a variable designated by an operand is shorter than a register of N-bit wide and the arithmetic operation instruction will possibly cause an overflow exceeding said effective bit-width.

21. The program converting unit of Claim 20, wherein said M is 16 and said N is an integer in a range of 17 to 31 inclusive.

24 ~~22.~~ The program converting unit of Claim 20, wherein said M is 32, and said N is an integer in a range of 33 to 63 inclusive.

25 ~~23~~. The program converting unit of Claim 20, wherein said compensate instruction generating means includes:

instruction judging means for judging an arithmetic operation instruction that will possibly cause an overflow for all the machine language instructions when said option instructing means holds the user's direction for executing the overflow compensation;

variable judging means, with respect to a variable in the arithmetic operation instruction judged by said instruction judging means, for judging an effective bit-width and whether said variable is signed or unsigned by referring to said table;

sign-extension instruction generating means for generating a compensation instruction in case of a signed variable, a logical value of a sign bit being filled into all bits higher than the effective bit-width in a register that is to store said signed variable by said sign-extension compensation instruction; and

zero-extension instruction generating means for generating a zero-extension compensation instruction in case of an unsigned variable, a logical value "0" being filled into all bits higher than the effective bit width in a register that is to store said unsigned variable by said zero-extension compensation instruction.

26 ~~24~~. The program converting unit of Claim 25 ~~23~~, wherein

said generating means includes:

determining ~~judging~~ means for determining ~~judging~~ a kind of the machine language instruction, the machine language instruction including (1) an instruction to access to an memory, (2) an instruction to use a register, and (3) an instruction to use an immediate;

memory managing means for outputting a direction, in case of the (1) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective memory-access width depending on the type of a variable to be accessed shown in said table;

register managing means for outputting a direction, in case of the (2) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective bit-width depending on the type of a variable to be read/written from/in the register shown in said table;

immediate managing means for outputting a direction, in case of the (3) instruction, to manage a corresponding bit-width held in said parameter holding means for the immediate as an effective bit-width depending on the type of the immediate shown in said table; and

code generating means for generating the machine language instruction in accordance with the directions from said memory managing means, said register managing means, and said immediate managing means.

22 ~~25~~. The program converting unit of Claim 21, wherein said M is 16 and said N is an integer in a range of 17 to 31 inclusive.

23 ~~26~~. The program converting unit of Claim 21, wherein said M is 32, and said N is an integer in a range of 33 to 63 inclusive.

Sub A4

27. A program converting unit for generating a machine language instruction based on a source program for a processor that manages an N-bit address while processing M-bit data, N being greater than M, said program converting unit comprising:

syntax analyzing means for analyzing a syntax of the source program to convert the same into an intermediary language comprising intermediary instructions, and subsequently for judging whether or not each variable contained in said intermediary instructions represents data used in an address;

table generating means for generating a table for each variable in said intermediary instructions, said table holding a name together with a type of each variable, said type representing one of the data and the address, and one of singed and unsigned data;

parameter holding means for holding a data width and a pointer width designated by a user, said data width representing the number of bits of the data while said pointer width representing the number of bits of the address;

114

option directing means for holding a user's direction for an overflow compensation, an overflow being possibly caused by an arithmetic operation;

generating means for generating an instruction to manage said data width when the variable in said intermediary instruction represents the data, and an instruction to manage said pointer width when said variable represents the address; and

compensate instruction generating means for generating a compensation instruction to compensate an overflow in accordance with a type of a variable used in the arithmetic operation, said type being judged when said option directing means holds the user's direction for executing the overflow compensation, said compensation instruction being generated when an effective bit-width of a variable designated by an operand is shorter than a register of N-bit wide and the arithmetic operation instruction will possibly cause an overflow exceeding said effective bit-width, wherein

said generating means includes:

judging means for judging a kind of the machine language instruction, the machine language instruction including (1) an instruction to access to an memory, (2) an instruction to use a register, and (3) an instruction to use an immediate;

memory managing means for outputting a direction, in case of the (1) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective memory-access width depending on the type of a variable to be accessed shown in said table;

register managing means for outputting a direction, in case of the (2) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective bit-width depending on the type of a variable to be read/written from/in the register shown in said table;

immediate managing means for outputting a direction, in case of the (3) instruction, to manage a corresponding bit-width held in said parameter holding means for the immediate as an effective bit-width depending on the type of the immediate shown in said table; and

code generating means for generating the machine language instruction in accordance with the directions from said memory managing means, said register managing means, and said immediate managing means, and wherein

said compensate instruction generating means includes:

instruction judging means for judging an arithmetic operation instruction that will possibly cause an overflow for all the machine language instructions when said option instructing means holds the user's direction for executing the overflow compensation;

variable judging means, with respect to a variable in the arithmetic operation instruction judged by said instruction judging means, for judging an effective bit-width and whether said variable is signed or unsigned by referring to said table;

sign-extension instruction generating means for generating a compensation instruction in case of a signed variable, a logical value of a sign bit being filled into all bits higher than the effective bit-width in a register that is to store said signed variable by said sign-extension compensation instruction; and

zero-extension instruction generating means for generating a zero-extension compensation instruction in case of an unsigned variable, a logical value "0" being filled into all bits higher than the effective bit width in a register that is to store said unsigned variable by said zero-extension compensation instruction.

28. A processor improved in address management comprising:

memory means for storing a program including an N-bit data arithmetic operation instruction and both N-bit and M-bit data load/store instructions, N being greater than M;

a program counter for holding an N-bit instruction address to output the same to said memory means;

fetching means for fetching an instruction from said memory means using the instruction address from said program counter; and

executing means for executing all N-bit arithmetic operation instructions and for executing N-bit and M-bit instructions excluding the arithmetic operation instructions, whereby an N-bit address is calculated by the N-bit arithmetic operation independently of a data bit-width, said data bit-width being M.

29. The processor of Claim 28, further comprising:

an address register group including a plurality of N-bit address registers;

a data register group including a plurality of N-bit data registers,

wherein said executing means executes the N-bit and M-bit data operation instructions using the address registers, while executing the M-bit data operation instruction using the data registers.

30. The processor of Claim 29, wherein :

said N is 24 and said M is 16; and

said processor is installed in a 1-chip microcomputer,

whereby said 1-chip microcomputer becomes suitable for running a program that utilizes a memory over 64 Kbyte for an operation with 16-bit data.

31. The processor of Claim 30, further comprising:

compensating means for extending an effective bit-width of the data in one of the address register and the data register to 24 bits,

wherein said compensating means operates in accordance with said compensate instruction entered ~~immediately~~ after a machine language instruction designating an arithmetic operation that will possibly cause an overflow.

32. The processor of Claim 31, wherein said compensating means includes:

a first extending unit for filling a logical value of a sign bit in all bits higher than the effective bit-width in a register;

a second extending unit for filling a logical value "0" in all bits higher than the effective bit-width in a register.

33. The processor of Claim 28, further comprising:

an address register group including a plurality of N-bit address registers; and

a data register group including a plurality of M-bit data registers,

wherein said executing means executes one of an N-bit data operation instruction and an M-bit data operation instruction using the address registers, while executing the M-bit data operation instruction using the data registers.

34. The processor of Claim 33, wherein :

said N is 24 and said M is 16; and

said processor is installed in a 1-chip microcomputer,

whereby said 1-chip microcomputer becomes suitable for running a program that utilizes a memory over 64 Kbyte for an operation with 16-bit data.

35. The processor of Claim 34, further comprising:

compensating means for extending an effective bit-width of the data in one of the address register and the data register to 24 bits,

wherein said compensating means operates in accordance with said compensate instruction entered ~~immediately~~ after a machine language instruction designating an arithmetic operation that will possibly cause an overflow.

36. The processor of Claim 35, wherein said compensating means includes:

a first extending unit for filling a logical value of a sign bit in all bits higher than the effective bit-width in a register;

a second extending unit for filling a logical value "0" in all bits higher than the effective bit-width in a register.

28
~~37~~. A processor for processing data in accordance with instructions in a program comprising:

register means including a plurality of register groups, each group being identical in bit-width while being different in types;

instruction decoding means for decoding an instruction to output register information indicating a register designated by an operand contained in a data-transfer instruction;

external-access-width control means for outputting the number of effective bits as bit-width information indicating a bit-width of transmission data in accordance with a kind of a register group to which said designated register belongs; and

external-access executing means for executing data transfer between said designated register and an external memory in accordance with said register information and said bit-width information.

29 ~~38~~. The processor of Claim 28 ~~37~~, wherein said register means includes:

an address register group including a plurality of address registers holding addresses; and

a data register group including a plurality of data registers holding data.

30 ~~39~~. The processor of Claim 29 ~~38~~, wherein

said external-access-width control means, as the bit-width information, outputs a bit width determined in accordance with the effective bit-width of the data used in the program when said register information represents the data registers, and outputs a bit-width determined in accordance with a sufficiently large address space for a program size and data area size of the program when said register information represents the address registers.

~~40~~ 31. The processor of Claim ~~38~~ 29, wherein:

the address registers and data registers in said register means are all 24-bit wide;

said instruction decoding means outputs information that represents one of the address register and the data register as the register information;

said external-access-width control means outputs the bit-width information exhibiting 24 bits when the register information representing the address register, and outputs the bit-width information exhibiting 16 bits when the register information representing the data register; and

the external-access executing means executes the data transfer three times and twice for the 24- and 16-bit-width information respectively for an 8-bit-width external memory, and for twice and once for the 24- and 16-bit-width information respectively for a 16-bit-width external memory.

~~41~~ 32. The processor of Claim ~~40~~ 31, wherein said access executing means includes:

an address generating circuit for holding an address designated by the data-transfer instruction to output one of a byte address and a word address to the external memory;

an output data buffer for holding write data designated by the data-transfer instruction to output the same one of per byte and per word to the external memory;

an input data buffer for holding data from read out from the external memory; and

a sequence circuit for outputting a byte address to said address generating circuit for an 8-bit-width external memory while controlling the number of times for the data-transfer in accordance with the bit-width information via the input/output data buffers with respect to the read/write data, for outputting a word address to said address generating circuit for a 16-bit-width external memory while controlling the number of times for the data-transfer in accordance with the bit-width information via the input/output data buffers with respect to the read/write data.

~~42~~ 33. The processor of Claim ~~38~~ 29, wherein:

the address registers and data registers in said register means are all 32-bit wide;

said instruction decoding means outputs register information indicating whether the instruction uses the address register or data register;

said external-access-width control means outputs the bit-width information exhibiting 24 bits when the register information representing the address register, and outputs the bit-width information exhibiting 16 bits when the register information representing the data register; and

the external-access executing means executes the data transfer three times and twice for the 24- and 16-bit-width information respectively for an 8-bit-width external memory, and for twice and once for the 24- and 16-bit-width information respectively for a 16-bit-width external memory.

34 ~~43~~. The processor of Claim 33 ~~42~~, wherein said access executing means includes:

an address generating circuit for holding an address designated by the data-transfer instruction to output one of a byte address and a word address to the external memory;

an output data buffer for holding write data designated by the data-transfer instruction to output the data one of per byte and per word to the external memory;

an input data buffer for holding data read out from the external memory; and

a sequence circuit for controlling said address generating circuit to output the byte address for an 8-bit-width external memory while controlling the input and output data buffers to input and output the byte data to transfer the read/write data to the external memory in a matching number of times to the bit-width of the external memory, and for controlling said address generating circuit to output the word address for a 16-bit-width external memory while controlling the input and output data buffers to input and output the word data to transfer the read/write data to the external memory in a matching number of times for the bit-width of the external memory.

Sub A5

44. A processor for operating certain data in accordance with an instruction in a program, comprising:

a first register means for holding N-bit data;

a second register means for holding N-bit data,

extending means for extending said M-bit data to N bits by copying an MSB of said M-bit data in a direction of an upper order, M being less than N;

zero-extending means for extending said M-bit data to N bits by copying a value "0" in a direction of an upper order;

operating means for operating an arithmetic operation in accordance with an instruction;

instruction control means for decoding an instruction to zero-extend M-bit immediate data when said M-bit immediate data are to be stored in said first register means by the decoded instruction and to sign-extend said M-bit immediate data when said M-bit immediate data are to be stored in said second register means by the decoded instruction, said zero-extended and sign-extended N-bit immediate data being outputted in one of two methods, one method being to send the extended N-bit immediate data from their respective extending means to their respective register means directly, the other being to send the same via the operating means to their respective register means.

IAC 8/20/97

~~45.~~ 36. The processor of Claim ~~44~~ 35, wherein

said first register means is a group of a plurality of address registers for storing addresses, and

said second register means is a group of a plurality of register means for storing data.

~~46.~~ 37. The processor of claim ~~45~~ 36, wherein said N is 24 and said M is 16.

Sub A6

47. A processor for operating certain data in accordance with an instruction in a program, comprising:

a first register means for holding N-bit data;

a second register means for holding N-bit data,

sign-extending means for extending said M-bit data to N bits by copying an MSB of said M-bit data in a direction of an upper order, M being less than N;

zero-extending means for extending said M-bit data to N bits by copying a value "0" in a direction of an upper order;

operating means for operating an arithmetic operation in accordance with an instruction;

instruction decoding means for decoding an instruction in the program to detect a first type instruction and a second type instruction, said first type instruction including an instruction to store M-bit immediate data into said first register means, said second type instruction including an instruction to store said M-bit immediate data into said second register means; and

control means for outputting said M-bit immediate data to said zero-extending means when the first type instruction is detected, and for outputting said M-bit immediate data to said sign-extending means when the second type instruction is detected, said zero-extended N-bit immediate data and sign-extended N-bit immediate data being outputted in one of two methods, one method being to send the extended N-bit immediate data from their respective extending means to their respective register means directly, the other being to send the same via the operating means to their respective register means.

39 ~~48~~. The processor of Claim 38 ~~47~~, wherein

said first register means is a group of a plurality of address registers for storing addresses, and

said second register means is a group of a plurality of register means for storing data.

40 ~~49~~. The processor of Claim 39 ~~48~~, wherein

said first type instruction includes a data-transfer instruction to store the M-bit immediate data to said first register means, an add instruction to add a value in said first register and the M-bit immediate data, and a subtract instruction to subtract the M-bit immediate data from a value in said first register, and said second type instruction includes a data-transfer instruction to store the M-bit immediate data to said second register means, an add instruction to add a value in said second register and the M-bit immediate data, and a subtract instruction to subtract the M-bit immediate data from a value in said second register.

41 ~~50~~. The processor of Claim 40 ~~49~~, wherein said N is 24 and said M is 16.

Sub A7

51. A data processing method for executing an instruction that includes an instruction to store M-bit immediate data in an N-bit first register and an N-bit second register, both M and N being integers while M being less than N, said method comprising the steps of:

decoding an instruction for selecting one of the first and second register in accordance with a decoded instruction;

zero-extending said M-bit immediate data to N bits when said decoded instruction designates the first register, and sign-extending said M-bit immediate data to N bits when said decoded instruction designates the second register; and

storing extended N-bit immediate data to the designated register.

43 ~~52~~. The method of Claim 42 ~~51~~, wherein

said first register means is a group of a plurality of address registers for storing addresses, and

said second register means is a group of a plurality of register means for storing data.

~~53~~ 44. The method of Claim ~~52~~ 43, wherein said N is 24 and said M is 16.

Sub A8

54. A processor for executing a program including an N-bit data arithmetic operation instruction, M-bit and N-bit load/store instruction, M being less than N, a conditional branch instruction, a data-transfer instruction with an external memory, and an instruction having immediate data, said processor comprising:

a first register means including a plurality of registers for holding N-bit data;

a second register means including a plurality of registers for holding N-bit data;

a program counter for holding an N-bit instruction address to output the same to said memory means;

fetching means for fetching an instruction from an external memory using the instruction address from said program counter;

instruction decoding means for decoding a fetched instruction;

executing means for executing all N-bit arithmetic operation instructions and for executing N-bit and M-bit instructions excluding the arithmetic operation instructions,

a plurality of flag storing means, each for storing a corresponding flag group changed in response to different bit-widths data in accordance with an execution result of said executing means;

flag selecting means for selecting a certain flag group from said plurality of flag storing means in accordance with a conditional branch instruction decoded by said instruction decoding means;

branch judging means for judging whether a branching is taken or not with a reference to a flag group selected by said flag selecting means;

sign-extending means for extending M-bit data to N bits by copying an MSB of said M-bit data in a higher order;

zero-extending means for extending M-bit data to N bits by filling a value "0" in a higher order;

compensation instruction control means for compensating contents of said first register means and said second register means using said sign-extending means and said zero-extending means in accordance with a compensation instruction inserted immediately after a machine language instruction for an arithmetic operation that will possibly cause an overflow, said machine language instruction being decoded by said instruction decoding means;

external-access-width control means for outputting bit-width information for transmission data in accordance with a type of said register means to which a register indicated by register information belongs, said register information indicating one of said first and second register means;

external-access executing means for executing a data transfer between the register and an external memory in accordance with said register information and bit-width information; and

immediate control means for outputting M-bit immediate data to said zero-extending means when a decoded instruction includes an instruction to store said M-bit immediate data in said first register means, and for outputting said M-bit immediate data to said sign-extending means when a decoded instruction includes an instruction to store said M-bit in said second register means, said zero-extended and sign-extended immediate data being sent to said first and second register means respectively in two methods, one being to send the same directly to their respective register means and the other being to send the same via said executing means.

~~55~~ 46. The processor of Claim ~~54~~ 45, wherein said N is 24 and said M is 16.

Add A9

125